2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS): GI 2016: 9th IEEE Global Internet Symposium

An Expectation-Based Approach to Policy-Based Security of the Border Gateway Protocol

Jun Li, Josh Stein, Mingwei Zhang
University of Oregon, USA
{lijun, jgs, mingwei}@cs.uoregon.edu

Olaf Maennel
Tallinn University of Technolog, Estonia
olaf@maennel.net

***Abstract*—The inter-domain routing protocol of the Internet, i.e., Border Gateway Protocol (BGP), is vulnerable to malicious attacks. Although many security solutions for BGP have been proposed, they have mainly focused on topology-based security. Policy-based security has largely been overlooked—a severe concern especially since BGP is a policy-based routing protocol. In this paper, we present an Expectation Exchange and Enforcement (E3) mechanism for defining policies between autonomous systems (ASes) such that any AS may enforce such policies.**

***Keywords*—Border Gateway Protocol (BGP); Policy-based routing security; Routing policy; Route leak**

I. Introduction

The Border Gateway Protocol (BGP) is the *de facto* standard inter-domain routing protocol on the Internet. By running BGP, every BGP router can determine its route toward an IP prefix anywhere on the Internet. To ensure BGP is secure against attacks, many BGP security solutions have been proposed. A good representation is BGPSEC [7]. However, while current BGP security solutions work well in many aspects, most BGP security solutions focus on topology-based security. They ensure that an attacker cannot impersonate the origin of an IP prefix, and/or that an attacker cannot insert itself onto the legitimate path for reaching an IP prefix. Typically, upon the receipt of a BGP update message, a BGP router checks that the update regarding an IP prefix does come from the true origin of the prefix, and/or that the update has traversed a particular path that consists of an ordered sequence of autonomous systems (ASes), i.e., AS path.

However, the current solutions seldom consider another aspect of BGP security—policy-based security. Specifically, they seldom consider whether the path conforms to routing policies of ASes *en route* or not, leaving BGP susceptible to attacks (or misconfigurations that violate routing policies), such as man-in-the-middle attack or route leak [2], [4], [9]. Figure 1 shows a simple route leak example. A route leak incident happens when a route is announced to neighboring ASes against its routing policy constraints, even if topologically the route is valid. Evidence has shown that this type of attack has become more than just ideas [12].

Given that BGP is a policy-based routing protocol, even though it is hard to gauge how many this type of attacks have occurred, the lack of attention toward security in the policy dimension is a significant concern. Figure 2 illustrates this problem. Assume a BGP security solution is in place and can enforce the routing security from topological perspective. In Figure 2a, if node 5 lies that it is or has a route to the origin of prefix p, any node can discover that this claim is fraudulent. However, if we now introduce an illegal route leak, the victims will not be able to capture it. In Figure 2b, assume node 3 should not be a transit node for traffic from node 1 or 2 to reach prefix p. However, upon receiving a routing update from node 4 ($4 \rightarrow D \rightarrow p$), node 3 propagates its route toward p ($3 \rightarrow 4 \rightarrow D \rightarrow p$) to nodes 2 and 1, even though doing so transgresses BGP policies. Nodes 1 and 2 are then likely to choose 3 as an intermediary hop for reaching prefix p, making it possible for node 3 to attract traffic it does not deserve. Note that although nodes 1 and 2 do run a BGP security solution, since the security solution only validates the topological correctness of routes, the updates they receive from node 3 will still look legitimate to them!

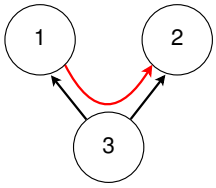


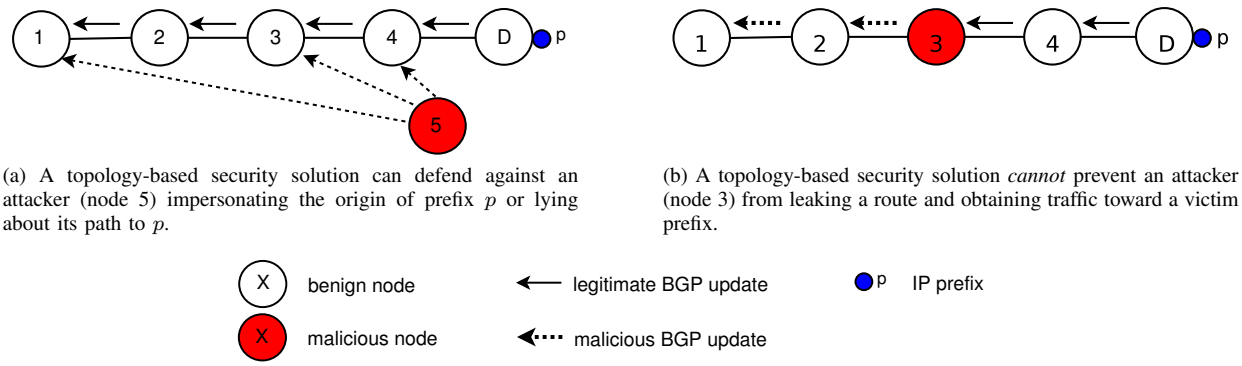
Figure 1: A route leak example. AS3 is a customer AS of AS1 and AS2, and it leaks to AS1 its route to AS2. AS1 thus learns a leaked route $1 \rightarrow 3 \rightarrow 2$ that AS1 should not use.

There are some existing policy-based security solutions. A well-known example is the enforcement of valley-free routing, in which for any AS along a route, either its previous hop, or its next hop, or both are customers of the AS in question. Valley-free routing is simple to implement by requiring each AS to announce their relationships [10]. However, due to most policy information being business secrets, most ASes will not reveal their relationships to non-participating parties, making it hard to distinguish between policy violations (such as route leaks) and normal operations. In fact, enforcing valley-free routing places an unnecessary restriction on ASes, and in practice the valley-free model is ignored [11].

Policy-based security is thus a critical problem yet to be fully addressed. We believe that, in addition to defining conventional routing policies, ASes should define policies to declare the legitimacy of routes, such as whether or not an AS can be included on a particular route, or what relationships be-

This material is based upon work supported by the USA National Science Foundation under Grant No. CNS-1118101. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

978-1-4673-9955-5/16/$31.00 ©2016 IEEE



(a) A topology-based security solution can defend against an attacker (node 5) impersonating the origin of prefix p or lying about its path to p.

(b) A topology-based security solution *cannot* prevent an attacker (node 3) from leaking a route and obtaining traffic toward a victim prefix.

Figure 2: The limitation of topology-based security solutions against policy-based routing attacks.

tween ASes along a route must be met. With route legitimacy defined by policies, we then are in a better position to close the security hole of ASes transiting traffic that they otherwise should not have access to.

We propose a policy-based security solution called Expectation Exchange and Enforcement, or E3, that allows for the enforcement of various routing policies between one or more ASes, in order that a newly advertised route meets policy expectations of ASes. It complements and is able to run alongside path integrity and origin authentication approaches such as BGPSEC. It can prevent damaging route leaks from polluting BGP routing tables and circumvent an AS from violating routing policies and placing itself as a man-in-the-middle into a path. Via E3, autonomous systems (ASes) are able to express routing policy expectations to other ASes (not necessarily only neighboring ASes but also other ASes as needed), allowing those ASes to validate routing policy compliance before incorporating a newly advertised path. In Figure 2a, if node 4 informs node 2 that node 4 expects to receive traffic from 2 via 3, then node 2 will accept the update from node 3. This could occur when, for example, node 3 is a legitimate backup provider of 4. Differently, in Figure 2b, if node 4 communicates to node 2 that it does not expect to receive traffic from node 2 via node 3, when route leak happens from node 3 to node 2, node 2 can detect the leak immediately and raise an alarm. For example, node 4 is a provider of node 3 and does not expect its customer to provide transit to other provider ASes.

II. Related Work

The policy-based BGP security approach that is taken in most works is the enforcement of valley-free routing [3]. HiBGP proposes an approach in which ASes explicitly state their relationship with another AS (e.g., provider), which lacks privacy and more importantly is coarse [10]. In paper [11], the authors propose a solution that involves a minimal flag embedded within a BGP update that is capable of maintaining the valley-free property without leaking relationship information. However, routing in the Internet, as the authors in [11] discovered, often do not follow valley-free routing.

Certain existing BGP security solutions can be leveraged for enforcing policy security to some extent. PGBGP is able to detect policy violations using the same approach it uses to detect anomalies in routing updates [6]. But after PGBGP detects a new link introduced by a policy violation, a human operator must determine the cause of the new route. IRV is capable of retrieving and enforcing policy information directly from ASes since it is able to act as a local Internet Routing Registry (IRR) [5]. However, no work has exploited this capability of IRV or defined what the policy would look like. Plus, the policies that are returned by IRV, and subsequently the IRR, are limited to the policies of that AS only. The policies for security purposes need to serve a wide range of situations; for example, a policy may be dictated through contracts between ASes, as can be seen in the design of E3 later.

A more general approach was taken in [16], in which the authors propose a method that allows for neighboring ASes to make promises concerning how their routing decisions are made. The promises are private in that policy information is not leaked yet the actual policy decisions that are made may be verified by the neighboring ASes. The limitation of this approach is that a degree of self-policing is present, by which an AS could feasibly make temporary promises that make certain attacks possible. It is therefore necessary that policies are dictated, at least partially, by ASes other than the primary AS in order to prevent an attacker from undermining the system. E3 instead incorporates the ability for defining general policies, while also requiring ASes to collaborate when defining inter-domain routing policies.

III. Expectation Exchange and Enforcement

A. Expectations

We designed an expectation exchange and enforcement mechanism, i.e., E3, to ensure that every advertised BGP route complies with BGP routing policies. We now define what an expectation is (including their types and their relationship), and how they are exchanged and enforced.

1) Definition of Expectations: The E3 mechanism at its core is composed of four concepts: expector, expectee, subject, and expectation. An **expector** is an AS that produces an expectation. An **expectee** is an AS that enforces an expectation. A **subject** is an AS that is specified in an expectation and directly affected by the expectation. An **expectation**, in its most basic form, is a set of IF-THEN rules, each stating if a condition is met, then what action will be taken. A condition can be a boolean function (such as whether a route contains

a specific link), or any standard relation (such as equal to, not equal to, less than, or greater than certain value) of any BGP attribute (such as a local preference value), or their AND/OR combinations. An action can be any of those defined in the Routing Policy Specification Language (RPSL) [1] (for instance, discarding a route). More formally, given an expector E, a subject S, and a set of IF-THEN rules each with condition C_i and action to take A_i, we can define an expectation as

$$e[E, S, C] = A, \quad where \quad C = \{C_i\}, \quad A = \{A_i\}.$$

It is important to note that the expectee is not present in the expectation, as any AS can be the expectee of this expectation to enforce it. We further categorize expectations into unilateral expectations, contractual expectations, and active expectations:

Unilateral Expectations: A unilateral expectation can be enforced immediately after declaration. However, unilateral expectations are prone to abuse by expector ASes as a malicious unilateral expectation can leave its subject AS as a victim.

Contractual Expectations: To prevent expectors from abusing the system, we require that an expector and its subject must construct a contractual expectation together. Usually defined for a long term, contractual expectations define the requirements that valid enforceable expectations must meet, including the boundaries for acceptable conditions and actions. Note an expectee does not directly apply a contractual expectation to filter advertised routes; it instead uses active expectations (discussed below).

Active Expectations: Active expectations are the expectations that are actively enforced. Active expectations, unlike contractual expectations, are produced solely by the expector. They are permitted to overlap and can be effective for only a short term, allowing an expector to define a set of expectations that satisfy their business needs as necessary. The caveat is that active expectations must be associated with a contractual expectation (although it is not required for a contractual expectation to have any active expectations).

2) Exchange of Expectations: In helping a BGP router to have up-to-date expectations, E3 supports two complementary modes of expectation exchange: the *query* mode and the *notification* mode. In the query mode, a BGP router queries specific ASes to learn their expectations. For example, if a BGP router runs PGBGP and receives a route that contains suspicious links, besides querying ASes that are the upstream endpoints of these links for topology-based security, it can also query those ASes about their expectations—especially if the router does not have expectations from them or only has old ones. In the notification mode, a BGP router notifies potential expectees of new expectations. Doing so, the potential expectees can receive new expectations—especially those of urgency—without delay. In particular, the router can record which ASes have requested its expectations recently, treat them as subscribers to its expectations, and update them immediately when new expectations become available.

We leverage the deployment of Resource Public Key Infrastructure (RPKI) [8], which is utilized in BGPSEC, for the purpose of authentication. Active expectations are signed by the expector whereas contracts, which require two parties, utilize more sophisticated multi-party signature schemes [14]. The authenticity provided by signatures also provides a means for expectations to be made tamper-proof.

3) Enforcement of Expectations: BGP updates must be checked against expectations to ensure routing policy compliance. We divide this task into two parts: checking a BGP update against active expectations, and checking an active expectation against its associated contractual expectation.

When checking if a BGP update message meets an active expectation, every IF-THEN rule of the expectation will be matched against the BGP update. If the condition of a rule is met, the action specified in the rule will be applied to the BGP update, such as those defined in the Routing Policy Specification Language (RPSL) [1]. For the purpose of protecting against illegitimate routes, dropping an advertised route is considered the default action.

We ensure that every active expectation, before we use it, is eligible by checking it against its associated contractual expectation. The criteria for eligibility are as follows:

1) All of the conditions in the active expectation must be a subset of the conditions in the contractual expectation.
2) The action of the active expectation must be the same as the action of the contractual expectation.

For example, if a contractual expectation is `CommunityValue < 200 → discard route` and an active expectation is `CommunityValue < 100 → discard route`, the active expectation is clearly a subset of the contractual expectation.

One particular issue that we handle is *conflicting expectations*. Sometimes an expector and a subject may have multiple contractual expectations between them, and if their conditions overlap, an active expectation may match more than one contractual expectation. Similarly, an expector may have multiple active expectations about a subject, and an expectee may be verifying a BGP route that matches more than one active expectation. We use a two tiered priority-based solution to address conflicting expectations. First, one expectation may be explicitly rated with a higher priority than another expectation, in this case the expectation with the higher priority is applied. If the expectations have the same priority, then any expectation that requires the route to be dropped will be chosen. On the other hand, if these overlapping contractual expectations demand the same actions, we can simply merge them into one expectation, so that an active expectation only maps to a single contractual expectation, or an update only maps to a single active expectation.

Figure 3 shows an example of enforcing expectations. In this example, AS2 receives two active expectations that involve AS3, with differing end goals of their expectors. In particular, AS1 is defining that valley routes are invalid, while AS4 is interested in a certain degree of preference. (The ability to define varying degrees of preference allows for an AS to define backup routes, such that they are unlikely to be chosen unless necessary.) The contractual expectations between AS3 and its two expectors, AS1 and AS4, would at minimum be supersets of the active expectations. For example, the contract for the first expectation may indicate `RouteContainsLink(1,3) OR CommunityValue(200) → discard route`.

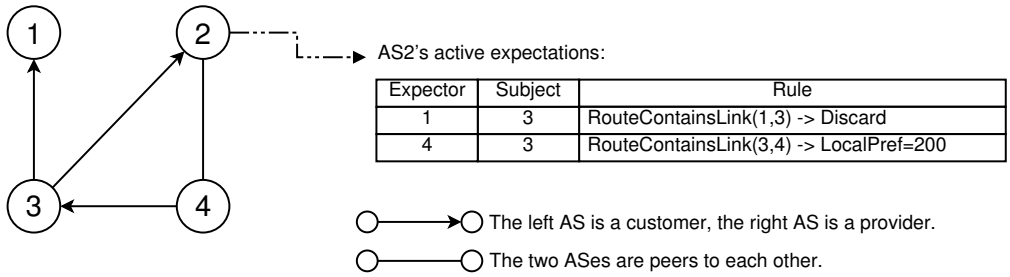


Figure 3: An example of expectation enforcement. AS2 has received multiple active expectations about AS3. The expectation from AS1 enforces valley-free routing, whereas the expectation from AS4 is designed to influence route selection by setting a local preference for routes that include AS3–AS4 link.

Upon receiving a BGP update, AS2 will check all expectations it received and determine which active expectations to apply. Assume AS2 received a route with AS-PATH <AS1→AS3>. AS2 will then apply its expectations, and discard this route (a valley route) according to the first expectation. Now, suppose that AS1 did not have a contractual expectation with the expector-subject pair (AS1, AS3), the first active expectation at AS2 will then be invalid, and AS2 will not apply the expectation to the route, meaning valley routes would be considered legitimate. Conversely, suppose that there was not an active expectation from AS1. Even if there is a contractual expectation indicating that AS1 has approval to forbid valley routes through AS3, the lack of an active expectation indicates that valley routes are still allowed, if only at that moment in time.

IV. Evaluation

A. Methodology

1) Metrics: We first measure when E3 is not deployed, what is the number of ASes that an invalid routing update can *pollute*—i.e., ASes that would accept and propagate invalid paths and use this number as a baseline. To evaluate the effectiveness of E3, we then use the baseline to measure the percentage of polluted ASes with different percentage of ASes that deploy E3. This metric allows us to evaluate the efficacy of E3 during incremental deployment.

2) Scenarios: A BGP route may be defined as either valley-free or as a valley. Additionally, the route may be considered valid, as in legitimate, or invalid. We thus may consider four scenarios in our evaluation:

- **Valley-Free, Valid**: A valley-free route is a route in which for any AS along the route, either its previous hop, or its next hop, or both are customers of the AS in question. Since valley-free routes satisfy the Gao-Rexford conditions [3], valley-free routes are considered valid and policy-compliant. For simplicity, all of our scenarios will consider routes that are valley-free as correct although E3 allows for the definition of valley routes.
- **Valley-Free, Invalid**: This scenario represents prefix hijacking events. Routes that are the result of a prefix hijacking, albeit valley-free, have long been considered invalid. Our evaluation of this scenario is limited to a case study of the 2008 Pakistan YouTube prefix hijacking event [13].
- **Valley, Valid**: Valley routes, in contrast to valley-free routes, have some AS whose previous and next hop are not customers, i.e. two peer neighbors (peer-peer valley) or two provider neighbors (provider-provider valley). Valley routes may be valid since they can represent complex business relationships such as backup providers to a multi-homed AS. Valley routes are common as indicated by [11] and although they portend that these are erroneous, the fact that routing is not disrupted by their presence indicates that valley routes are typically benign. Although it is well within the capability of E3 to address it, we do not explore this scenario in our evaluation.
- **Valley, Invalid**: This scenario represents the current stance toward valley routes. For simplicity, our evaluation also takes the stance that valley routes are invalid although this is a narrow application of E3. We evaluate E3 in this scenario using both simulations and a case study of the 2012 Canada route leak event [2].

Due to the space limitation, our evaluation looks into the *Valley-Free, Valid* and *Valley, Invalid* scenarios. The impact of deploying E3 in other scenarios will be our future work.

3) Autonomous Systems (ASes): We classify ASes according to their AS rank. We classify the first 100 as tier 1 and the next 900 as tier 2. We then deploy E3 over varying percentages of tier-1 or tier-2 ASes.

B. Simulations

We focus our investigation on route leaks that violate the valley-free model, i.e., peer-peer valley and provider-provider valley. From our deployment model, we deploy E3 on a percentage of tier 1 ASes. Each percentage of deployment consisted of 100 simulations, in which 10 ASes that are known to produce valley routes would leak a route for 10 deployments of a given configuration. We then measured the number of ASes that chose the leaked route as their best path.

Figure 4 shows the results. In the worst case in which no ASes are deploying E3, there were approximately 200 ASes that chose the invalid, peer-peer valley route as their best path, and approximately 5000 ASes that chose the invalid, provider-provider valley route as their best path. When 100% tier-1 ASes deploys E3, nearly 80% of ASes originally polluted are then protected from invalid valley routes. A notable feature of this scenario is the lack of effectiveness except when E3 is deployed on a relatively high percentage of tier-1 ASes.

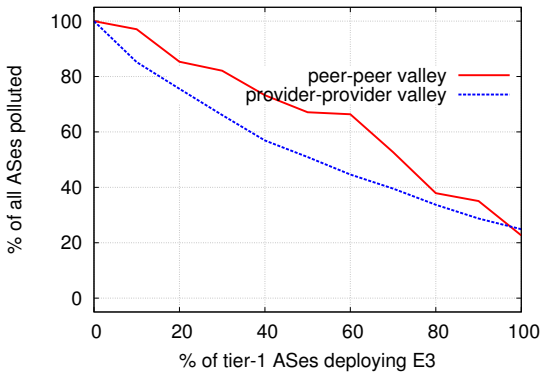


Figure 4: E3 deployment simulation for peer-peer and provider-provider valleys.

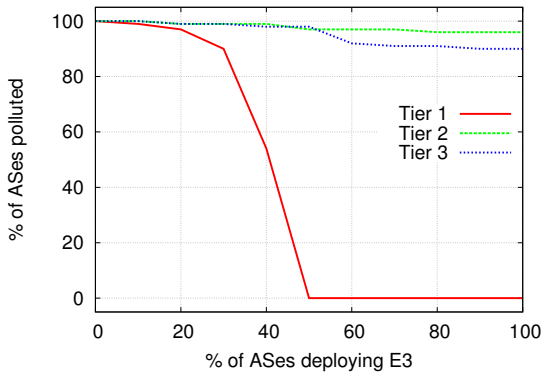


Figure 5: E3 case study on 2012 Canada route leak event.

This phenomenon is related to where a route leak occurs. If a route-leaking update does not traverse an E3-enabled tier-1 AS, which is often the case, the route will not be prevented from propagating further. (We further analyze the deployment strategy of E3 in Section IV-D.) Also, provider links appear more often over routes on the Internet in comparison to peer links, particularly due to the preference of a provider choosing routes from their customers over their peers. There are therefore more opportunities for provider-provider valley routes to be prevented.

C. Real-world Cases

In this section, we evaluate the effectiveness of E3's prevention mechanism in real-world cases. We use the 2012 Canada route leak event [2] as an example case to show how E3 can prevent the propagation of large-scale route leak anomalies. The route leak occurred on August 8, 2012, when a Canadian ISP leaked its full routing table to its provider. Unlike the China Telecom route leak [15], where China Telecom claimed to be the origin of many of the affected prefixes, this event was more subtle; the leaker simply claimed to be on the path to the affected prefixes. During the event, the Canadian ISP Dery Telecom Inc (AS 46618) leaked all its routes acquired from one of its provider VideoTron (AS 5769) to its another provider Bell (AS 577). Bell selected a large portion of these routes as best routes and then further propagated to its peers. This route leak event affected 107,409 prefixes from 14,391 different ASes across the Internet [2].

We investigated the efficacy of E3 with different levels of deployment. When E3 is not deployed, there would be a total of 47201 invalid propagations. We then randomly chose a certain percentage of affected tier-1, tier-2, or tier-3 ASes to deploy E3, and we analyzed the propagation of invalid route-leak routes after the deployment. For each combination of tier and percentage, we ran the experiments 1000 times to extract the mean number of propagations of invalid updates. Our analysis shows that the route leaks usually have bottle-neck ASes that determine the propagation scope; deploying E3 on these bottle-neck ASes can most effectively prevent the route leak; and deployment on tier-1 ASes has the best effectiveness. As shown in Figure 5, with 50% E3 deployment on the affected tier-1 ASes, E3 could prevent almost 100% of invalid propagations.

D. Deployment Strategy Analysis

We also analyzed deployment strategies of E3. We took the affected paths from our previous studies and examined the ASes that appeared after the leaking AS, i.e., the AS that is responsible for leaking a valley route. The ASes we examined are candidate deployment locations of E3, which we also call **tail ASes**. In order to forecast the preventative capabilities of E3 given a particular deployment strategy, we looked at the distribution of tail ASes by tier that appear X hops after the leaking AS.

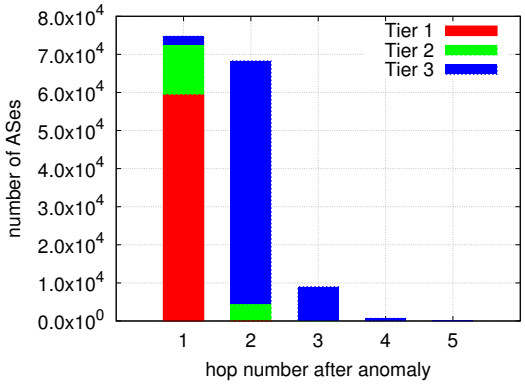
The results of our analysis under 0% E3 deployment are shown in Figures 6a and 6b. There are two clear differences between the two figures: the path lengths, and the occurrences of ASes by tier for a given hop. Path lengths involving a provider-provider valley can be over twice as long as those involving a peer-peer valley. This result is reasonable considering that provider-provider valleys involve a customer link which has a higher preference over a peer link. The higher preference clearly translates to greater acceptance of the route across the Internet. The second observation provides more utility for the purposes of effective deployment.

The peer-peer valley scenario, shown in Figure 6a, hints that tier-1 ASes would be the most strategic ASes for preventing such a route leak, since tier-1 ASes make up the majority the ASes after the leaking AS. The provider-provider scenario, shown in Figure 6b, shows tier-1 ASes at nearly every hop after the leaking AS, thus tier-1 ASes would also be effective deployment ASes for preventing provider-provider route leaks.

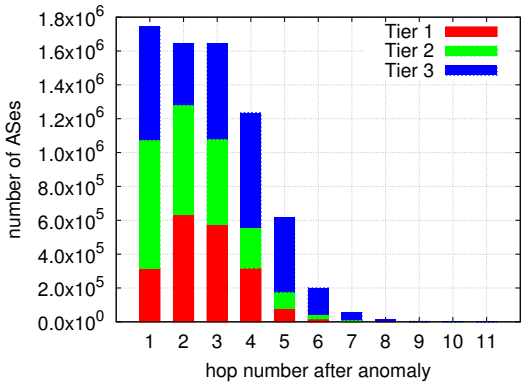
V. Discussion

We have shown that E3 provides preventative capabilities in both simulated and real-world scenarios, particularly in the case of route leaks. However, several important research issues remain unaddressed.

One open issue facing E3 is its implementation. E3 could be implemented and running on every BGP router (like BGPsec), or on a dedicated server at every AS where the server is further connected to the BGP routers in the AS (like IRV).



(a) Occurrences of ASes by tier and hop distance from the leaking AS under a Peer-Peer valley scenario.



(b) Occurrences of ASes by tier and hop distance from the leaking AS under a Provider-Provider valley scenario.

Figure 6: Tail AS analysis.

The former would lead to an in-band approach in that E3 expectations will be added to BGP update messages, while the latter would lead to an out-of-band approach as it keeps BGP updates intact and relies on out-of-band channels for ASes to communicate E3 expectations. Also, expectation, in its current form, is an abstract concept with well-defined properties. Expectations therefore require a format, such as RPSL, so that they may be utilized by ASes [1].

Another open issue is how an AS is to choose between E3's query mode and notification mode. The query mode is primarily a pull-based system, which introduces latency between expectations being published and the expectations being enforced. This latency, regardless of its length, is a gap in security that could be exploited. The notification mode incurs little latency, but it could result in high overhead as the number of possible subscribers to an AS' expectations could be as many as all ASes.

Moreover, E3 still requires further evaluation. For example, it will be useful to obtain E3's efficacy under other scenarios described in Section IV-A2. It would also be useful to know the additional communication overhead that E3 adds. It is perhaps even more important to further study the incentives and strategies for deploying E3, such as the cost-benefit trade-off in running E3. Finally, while E3 complements those topology-based BGP security solutions, the cost and performance when it works together with these solutions needs to be better understood.

VI. Conclusion

Most of BGP security approaches have focused on securing routes such that they are topologically valid. The possibility that topologically valid routes may be still illegitimate and violate routing policies has been largely overlooked. As BGP is a policy-based routing protocol, the lack of a systematic policy-based security solution must be addressed. In this paper, we introduced Expectation Exchange and Enforcement (E3), a BGP extension for expressing and enforcing policies across ASes to prevent policy-violating routes from propagating. E3 defines expectation in a general format, supports the secure exchange of expectation between ASes, and enables ASes to enforce security policies conveyed in expectations. Our evaluation shows that E3 is effective, especially when deployed by a decent percentage of tier-1 ASes. Future work of E3 includes its implementation, expectation exchange mode selection, and more evaluations.

References

[1] Routing Policy Specification Language. http://www.irr.net/docs/rpsl.html.

[2] "BGPMon". A BGP leak made in canada, 2012.

[3] L. Gao, T. G. Griffin, and J. Rexford. Inherently safe backup routing with bgp. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 1, pages 547–556. IEEE, 2001.

[4] S. Goldberg, M. Schapira, P. Hummon, and J. Rexford. How secure are secure interdomain routing protocols? In *ACM SIGCOM*, 2010.

[5] G. Goodell, W. Aiello, T. Griffin, J. Ioannidis, P. McDaniel, and A. Rubin. Working around BGP: An incremental approach to improving security and accuracy of interdomain routing. In *10th Network and Distributed System Security Symposium (NDSS)*, volume 3, pages 75–85, February 2003.

[6] J. Karlin, S. Forrest, and J. Rexford. Autonomous security for autonomous systems. *Computer Networks*, 52(15):2908–2923, 2008.

[7] M. Lepinski, R. Austein, S. Bellovin, R. Bush, R. Housley, S. Kent, W. Kumari, D. Montgomery, K. Sriram, and S. Weiler. BGPSEC Protocol Specification. http://tools.ietf.org/html/draft-ietf-sidr-bgpsec-protocol-06, May 2012. work in progress.

[8] M. Lepinski and S. Kent. An Infrastructure to Support Secure Internet Routing. RFC 6480, February 2012.

[9] D. McPherson, S. Amante, and E. Osterweil. Route Leaks & MITM Attacks Against BGPSEC, draft-ietf-grow-simple-leak-attack-bgpsec-no-help-02. IETF Internet Draft, November 2012.

[10] J. Qiu and L. Gao. Hi-BGP: A lightweight hijack-proof inter-domain routing protocol. *University of Massachusetts Amherst Technical Report*, 2006.

[11] S. Qiu, P. McDaniel, and F. Monrose. Toward valley-free inter-domain routing. In *IEEE International Conference on Communications*, pages 2009–2016, June 2007.

[12] "Renesys". The new threat: Targeted internet traffic misdirection, 2013. http://www.renesys.com/2013/11/mitm-internet-hijacking/.

[13] RIPE NCC. YouTube hijacking: A RIPE NCC RIS case study. http://www.ripe.net/news/study-youtube-hijacking.html.

[14] D. Tonien, W. Susilo, and R. Safavi-Naini. Multi-party concurrent signatures. In *Information Security*, volume 4176, pages 131–145. 2006.

[15] A. Toonk. Chinese ISP hijacks the internet. http://bgpmon.net/blog/?p=282, April 2010.

[16] M. Zhao, W. Zhou, A. Gurney, A. Haeberlen, M. Sherr, and B. Loo. Private and verifiable interdomain routing decisions. In *Proceedings of the ACM SIGCOMM*, pages 383–394, August 2012.